UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

Diarización de Locutores en Audio Broadcast

Gonzalo Soriano Morancho
Tutor: Joaquín González Rodríguez

Mayo 2016

DIARIZACIÓN DE LOCUTORES EN AUDIO BROADCAST

AUTOR: Gonzalo Soriano Morancho
TUTOR: Joaquín González Rodríguez

Biometric Recognition Group - ATVS
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Mayo de 2016

Resumen

La diarización de locutores es un campo poco explorado dentro del análisis de la voz y, por tanto, tiene ahora mismo un gran interés en investigación.

El desarrollo de este trabajo se centrará en el análisis del estado actual de este campo, concretamente, nos centraremos en el sistema LIUM. Un sistema opensource que tuvo muy buenos resultados en las evaluaciones ESTER 2, ETAPE y REPERE.

En primer lugar estudiaremos este sistema en base a pruebas con la evaluación de Albayzin 2010, con el objetivo de tener una idea concreta del funcionamiento del sistema. Más tarde evaluaremos este sistema para una base de datos desarrollada por un grupo de estudiantes de la universidad. Dicha base de datos contiene programas de radio en difusión actualmente en España.

La última etapa de este Trabajo será el adaptar de la manera más ajustada posible este sistema a nuestra base de datos, detallando a su vez las etapas más importantes y, por tanto, en las que más esfuerzo computacional hay que invertir.

Esta última etapa ha consistido en el estudio de una gran cantidad de datos provenientes de simulaciones con parámetros ajustados de manera manual del sistema estudiado.

Palabras clave

Audio, reconocimiento, voz, procesado, broadcast, diarización, algoritmo, clustering, radio, base de datos, tramas

Abstract

Audio Diarization is a field of study which has become unstudied over the last years, therefore it has an undeniable interest on nowadays investigation.

This Bacherlor Thesis will take over the current state of the art in this field, we will mostly study an existing system which has been probed to work reasonably well. This system from Le Mans' University, called LIUM, is an open source system which has become winner on ESTER 2, ETAPE and REPERE evaluations.

Firstly we will study the system using the Albayzin 2010 benchmark, this will provide us with the general system's usage, performance and reliability. Lately we will evaluate this system with a database which will be created on collaboration with other university students, at the beginning of this Thesis. This database will be created with a group of radio shows broadcasted currently on Spain.

The last step will be to adapt, in the finest way possible, this system to our new database, in order to adapt its performance to the type of corpus we will want to study. This will also provide us with the opportunity to study each of the components of the system separately, so we will know which one needs the most effort to be improved.

This last step consists on the study of a large amount of data coming from different simulations with different input parameters which will be adjusted manually to the system under study.

Keywords

Audio, voice, recognising, processing, broadcast, diarization, algorithm, clustering, radio, database, frames.

Agradecimientos

En primer lugar quiero agradecer a todos los integrantes del grupo ATVS su ayuda inestimable, tanto en la creación de la base de datos como en sus posteriores aportaciones en cualquier problema surgido. En especial a Cristian Sánchez, por su guiado mientras realizaba su TFM, y a Joaquín González, por su guiado y por proveerme de las herramientas, no solo para este TFG, sino también para seguir con estudios sobre esta materia más adelante.

Por otro lado, gracias a mi familia, especialmente a mi madre, por haberme apoyado durante toda mi vida y por sus actitudes, que siempre han sido, y serán, un ejemplo para mí.

A mis amigos, que tan bien han sabido darme una palabra de aliento cuando era necesario y que me han acompañado en esta etapa que, ahora, llega a su fin.

A mi padre y a mi abuela.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 Introducción

1.1 Motivación

Pese a que el procesado de voz es un campo que goza de gran interés investigador, y más ahora en esta época de asistentes virtuales y demás diversos sistemas de reconocimiento del habla, la separación de locutores en una conversación no ha sido un campo muy investigado.

Los reconocedores de voz clásicos basan su funcionamiento en el reconocimiento de lo que se está diciendo, esto es, responden a la pregunta de ¿qué se está diciendo? Por otro lado, los sistemas de reconocimiento de hablantes intentan responder a la pregunta ¿quién está hablando? Sin embargo estos sistemas, sobre todo el reconocedor de hablantes, precisan de gran cantidad de datos para devolver un resultado fiable.

Los sistemas de diarización pues tratan de resolver las preguntas ¿quién habló y cuando habló? Así pues pueden tener un parecido conceptual a un reconocedor de hablante, pero utilizando segmentos muy pequeños de audio que han de ser evaluados.

Por esta razón, la diarización de locutores es una etapa de preprocesado muy interesante de cara a sistemas de reconocimiento del hablante. El sistema estaría encargado así de anotar los diferentes tiempos en que un hablante en particular ha intervenido y, juntando los segmentos de un mismo hablante, este ya puede ser estudiado más en profundidad por el sistema de reconocimiento. Así pues también es útil de cara a sistemas de detección automática del habla ya que puede mejorar la calidad de la transcripción por permitir el uso de un sistema de reconocimiento del habla adaptado a cada locutor individualmente.

1.2 Objetivos

El objetivo principal de este proyecto es el de abrir una nueva línea de investigación de la universidad que analice y proponga soluciones a estos problemas descritos anteriormente.

El primer paso para la consecución de este objetivo pues, es el de analizar con profundidad las herramientas existentes del estado del arte, comprender su funcionamiento, analizar sus bondades y sus flaquezas para poder entender, con la profundidad requerida, los siguientes pasos a realizar en este campo.

De manera secundaria, se ha creado una base de datos que permita comprobar, en un entorno controlado, el funcionamiento del sistema en cuestión. Esta base de datos de audios de radio española ha sido recopilada y etiquetada por miembros del ATVS Biometric Group, y presenta distintos entornos en cuanto a calidad de grabación, ruidos de fondo, entornos de grabación, etc.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

Capítulo 1. Introducción.

En este primer capítulo se explican las motivaciones, objetivos perseguidos. Se realiza también una explicación de la estructura que tiene el trabajo.

Capítulo 2: Estado del arte.

A lo largo de este capítulo se realiza un estudio del estado actual de la tecnología bajo estudio. Se dará una visión del funcionamiento de los sistemas de diarización, centrándonos en sistema bajo estudio. Estudiaremos sus características y componentes más importantes.

Capítulo 3. Entorno experimental.

Se describe el proceso mediante el cual se ha adaptado el sistema bajo estudio a nuestro marco de trabajo. Además se presentan la metodología para la obtención de resultados.

Capítulo 4. Pruebas y resultados.

Se detallan las pruebas llevadas a cabo y se estudian los resultados obtenidos.

Capítulo 5. Conclusiones y trabajo futuro.

Se hace una valoración final del resultado final del algoritmo y se detallan posibles líneas de investigación futuras.

2 Estado del arte

La diarización de locutores es el proceso para responder a la pregunta de ¿quién habló y cuándo? También se le puede denominar como segmentación y clustering de locutores. Este otro nombre proviene de que es el proceso de segmentación del audio y agrupamiento de acuerdo a la identidad del hablante. Sin embargo, es importante tener en cuenta que, contrariamente al reconocimiento de locutores, en esta tarea no se busca la identidad real del hablante, este será denominado simplemente como spk1, spk2, etc.

2.1 Resumen del sistema

Fundamentalmente hay dos aproximaciones típicamente utilizadas en los sistemas de diarización: *bottom-up* y la *top-down*.

En primer lugar la *bottom-up*, usada en el sistema que estudiaremos, es también conocida como *Agglomerative Hierarchical Clustering (AHC).* La idea de esta aproximación es que en un primer momento el audio es segmentado en trozos demasiado pequeños *(oversegmented)* y, por lo tanto, el número de segmentos es mayor al número de hablantes. Esta segmentación representa, por tanto, la primera etapa del proceso de clustering. Los clústeres entonces son unidos de manera gradual utilizando criterios de similitud entre ellos utilizando un criterio de parada cuando se obtengan el número correcto de hablantes.

Por otro lado, la aproximación *top-down*, utiliza una estrategia opuesta. En este caso se utiliza un clustering de división que, comenzando con un número de pequeño de clústeres iniciales, va dividiendo los mismos de acuerdo a un criterio de similitud. Este proceso de división de clústeres es detenido, idealmente, cuando se llega al número indicado de hablantes.

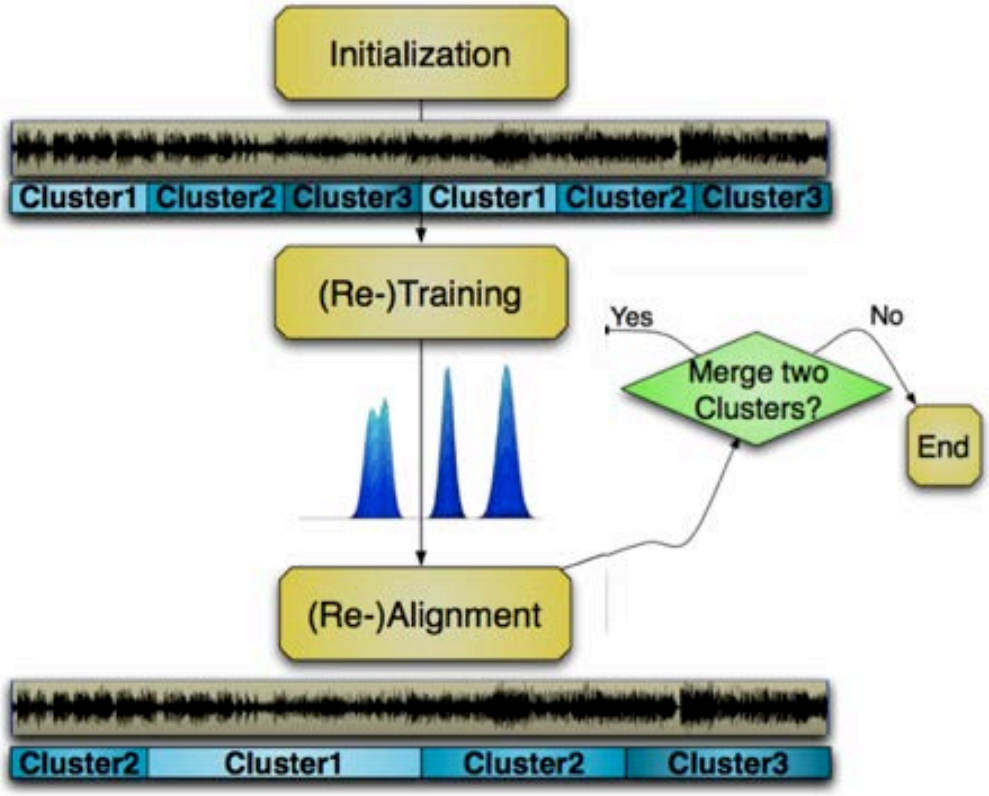

Figura 2-1: Algoritmo de clustering AHC. Extraída de [2]

2.2 Extracción de características

Cualquier hablante tiene, en su forma de hablar, características que hacen su voz única. Estas características tienen su origen principalmente en la fisiología del tracto vocal de cada persona, así como en la forma de articular esa fisiología concreta. Por este motivo, la primera tarea de cualquier sistema de procesamiento de voz es el de cuantificar estas características. En el análisis de voz uno de los métodos más habituales es el de los Mel-Frequency Cepstral Coefficients (MFCCs).

El proceso de extracción de los MFCCs sigue los siguientes pasos:

1. La señal de audio es enventanada. Normalmente se utiliza una ventana de tipo Hamming de 20 ms de duración con un 50% de solape.

2. Se aplica la Transformada Rápida de Fourier (FFT) a cada ventana.

3. Se hace pasar la señal por un banco de filtros en escala de frecuencias de Mel.

4. Se calcula el logaritmo de la energía de la salida de cada filtro.

5. Por último, se aplica la Transformada Discreta del Coseno (DCT). Debido a que esta transformada compacta la mayoría de la energía en los primeros coeficientes, no es necesario obtener una gran cantidad de ellos, típicamente se obtienen los trece primeros.

2.3 Segmentación

La segmentación es un bloque fundamental en cualquier sistema de diarización. Si la entrada a esta etapa es una señal de audio sin segmentar, intenta llevar a cabo la separación entre segmentos de voz/no voz, a la vez que intenta detectar los puntos de cambio de locutor. En cambio, si la entrada es la salida de un sistema de *speech activity detection*, su objetivo será el dividir la parte del audio clasificada como voz en segmentos de cada hablante.

La aproximación más habitual a este paso consiste en una comprobación de una hipótesis para la similitud de los segmentos en dos ventanas solapadas consecutivas. Normalmente el deslizamiento de una ventana sobre otra se produce en pasos de 100 ms.

Para cada posición del deslizamiento de la ventana hay dos posibles hipótesis: primera, hay dos hablantes distintos y, por tanto dos modelos diferentes es más apropiado; y, segunda, los dos segmentos de audio están bien representados por un solo modelo.

En la práctica, los modelos son calculados para cada ventana de análisis y algún criterio conocido a priori se usa para determinar si son mejor considerados por dos modelos separados o por un solo modelo. Los modelos son representados normalmente por distribuciones gaussianas.

Hay multitud de medidas de distancia (similitud entre ventanas). Una forma muy utilizada es la *Bayesian Information Criteria (BIC)* y su asociado *ΔBIC*, que calcula la distancia entre dos clústeres para decidir si son modelados de mejor manera por un solo modelo (no hay punto de cambio de audio) o por dos modelos (hay punto de cambio). La decisión para saber si unir dos clústeres c_i y c_j se calcula *ΔBIC* como:

$$\Delta BIC(c_i||c_j) = (n_i + n_j)\ln|\Sigma| - n_i \ln|\Sigma_i| - n_j \ln|\Sigma_j| - \lambda P$$

donde Σ es la matriz de covarianza del clúster unificado (c_i y c_j), Σ_i del clúster c_i, Σ_j del clúster c_j; y los n_i y n_j son el número de ventanas en los respectivos clusters. La penalización se define como:

$$P = \frac{1}{2}\left(d + \frac{1}{2}d(d+1)\right)\ln n$$

donde *d* es la dimensión de vector de características y $n = n_i + n_j$.

2.4 Clustering

La salida del bloque de segmentación puede ser considerada como el inicio de la etapa de clustering en la que cada clúster es representado por un solo segmento.

Esta etapa, basada en la aproximación *bottom-up*, utiliza AHC, que puede ser dividido en las siguientes etapas:

1. Calcular distancias por pares para cada cluster.

2. Unir los clústeres más próximos.

3. Recalcular las distancias con los clústeres pendientes.

4. Rehacer todos los pasos anteriores hasta que el criterio de parada se cumpla.

Los clústeres en cada iteración son recalculados y representados por una matriz de covarianza gaussiana. Aunque también puede ser modelada por una matriz diagonal de covarianza gaussiana.

El principal objetivo de esta etapa es la de unir los segmentos correspondientes al mismo hablante. Mientras que el clustering de la etapa de segmentación opera en ventanas consecutivas para determinar si corresponden o no al mismo hablante, el clustering de esta etapa trata de identificar y unir los segmentos de un mismo hablante sea cual sea su posición en el audio.

2.5 Realineamiento Viterbi

Tras el clustering, es habitual ejecutar el algoritmo de Viterbi para realinear los extremos de los distintos segmentos. Este proceso ayuda a reducir el DER final del sistema.

Para realizar esta tarea se realiza en primer lugar un Hidden Markov Model (HMM) que deberá ser entrenado con la salida del bloque de clustering.

El HMM tendrá tantos estados como número de clústeres se haya generado en el bloque de clustering, siendo representado cada uno por un Gaussian Mixture Model (GMM) entrenado con los datos de cada cluster.

En el HMM la secuencia generada por cada estado es una serie de conjuntos de características que representan la señal de audio.

Si el HMM contiene un número m de estados que generan n conjuntos de características el HMM será representado por:

- Una matriz MxM que contendrá la probabilidad de transición de cada uno de los M estados a cada siguiente estado, incluyéndose a sí mismo.

- Una matriz MxN que contendrá la probabilidad de cada conjunto de características para cada uno de los clústeres.

Juntando toda esta información el realineamiento puede ser realizado con el algoritmo de Viberbi, que se encarga de estimar la secuencia de estados más probable que represente la secuencia de conjuntos de características dadas.

Para nuestro caso la ecuación correspondiente para el cálculo del modelo GMM para una dimensión n, cada distribución sería:

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

donde μ es el vector n-dimensional de la media y Σ la matriz de covarianza de tamaño nxn.

El GMM está definido por la superposición de K distribuciones como la enunciada, de la forma:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k,\Sigma_k)$$

donde cada componente $\mathcal{N}(x|\mu_k,\Sigma_k)$ tiene su componente de media y covarianza. Los parámetros π_k representan la influencia de cada componente en el GMM, y deben cumplir:

$$\sum_{k=1}^{K} \pi_k = 1$$

2.6 Clustering SID

El proceso de clustering AHC descrito en el punto 2.4 tiene que trabajar con segmentos de duración muy corta, por lo que el número de parámetros para modelar cada clúster en las primeras iteraciones es escaso. Tras realizar algunas iteraciones la cantidad de datos por clúster aumenta, por lo que es posible aplicar un modelo más complejo.

La hipótesis manejada es la de detener el AHC en una etapa temprana, evitando así el *underclustering*, es decir, detenemos el AHC cuando el número de clústeres es todavía mayor al número de hablantes. De hecho, en el caso de que no se hayan aplicados métodos para la normalización de canal en una etapa previa al AHC, un solo hablante aparecerá

como varios clústeres debido a las cambiantes condiciones del entorno en el que se encuentra. Por tanto, es necesario aplicar técnicas de normalización de canal e, incluso, la dimensión de los conjuntos de características son ampliadas con los coeficientes Δ y $\Delta\Delta$.

Esta técnica de clustering requiere el uso de Universal Background Model (UBM). Un UBM es un modelo GMM grande, en nuestro sistema de 128, diseñados para representar la distribución de características de todos los hablantes en general. Básicamente normalmente se utilizan dos UBM, uno que distingue el sexo del hablante (masculino o femenino) y otro que distingue entre voz telefónica o voz de estudio. El método que combina el GMM con el UBM procesa los clústeres resultantes de la etapa de clustering AHC, cada uno modelado con un GMM.

El criterio de agrupamiento para el clustering suele ser el Cross Likelihood Ratio (CLR) y, cuando este toma cierto valor, también es el cálculo utilizado para detener el agrupamiento. Es también muy utilizado este valor normalizado (NCLR) ya que ha demostrado mejor rendimiento en muchos casos:

$$CLR\ (c_i||c_j) = \ln\frac{L(x_i|M_j)}{L(x_i|M_{UBM})} + \ln\frac{L(x_j|M_i)}{L(x_i|M_{UBM})}$$

$$NCLR\ (c_i||c_j) = \frac{1}{n_i}\ln\frac{L(x_i|M_j)}{L(x_i|M_{UBM})} + \frac{1}{n_j}\ln\frac{L(x_j|M_i)}{L(x_i|M_{UBM})}$$

donde M_i y M_j son los GMM de los clústeres c_i y c_j después de ser pasados por un algoritmo de *Maximum a posteriori adaptation,* cuyos detalles se pueden estudiar en [3]. n_i y n_j son el número de conjuntos de características en los clústeres. $L(x_i|M_j)$ y $L(x_i|M_{UBM})$ son la probabilidad del vector de características x_i, representando a c_i, dado el modelo M_j o M_{UBM}.

En cada iteración de este algoritmo, cada par de clústeres con la CLR, o NCLR dependiendo del sistema concreto, más altas son juntados en un solo clúster y un nuevo modelo es calculado. Este proceso es recurrente con un umbral de parada previamente definido.

2.7 Speech Activity Detection

Cualquier sistema de diarización ha de trabajar con los segmentos de audio que contengan voz para ser diarizada. Es por esto que, un correcto etiquetado del habla con el Speech Activity Detection (SAD) es de capital importancia para el rendimiento del sistema y, además, es responsable directo de los errores de False Alarm -diarizar cuando no hay voz- y Missed Speech –no diarizar voz cuando sí que la hay-. Incluso, podría llegar el caso de que un incorrecto etiquetado voz/no voz, afecte al proceso de diarización introduciendo datos incorrectos en el cálculo de los modelos estadísticos, lo que haría incrementar el Speaker Error –asignar un segmento de voz a un hablante equivocado-.

Un problema importante a tener en cuenta es que, en el caso del audio broadcast, los segmentos sin voz suelen no ser de silencio, aunque pueden serlo, además hay música, aplausos, ruidos de fondo, etc. E incluso estos ruidos se pueden solapar unos con otros, dando lugar a unas mezclas que hacen que el desarrollo del SAD sea complejo.

Es muy habitual seguir la estrategia de Voice Activity Detector (VAD) que cuentan con un umbral de energía a partir del cual se considera un segmento de voz. Es una solución simple, pero que no tiene en cuenta la casuística descrita ya que, por ejemplo, un segmento con música podría superar ese umbral.

La estrategia de LIUM [4] es más efectiva, pero también más compleja, consiste en el uso de modelos que caractericen cada posible estado. En el caso concreto de LIUM el audio se asigna a ocho categorías distintas: 2 modelos para silencio –de banda ancha y de banda estrecha-, 3 modelos para voz de banda ancha –voz limpia, voz solapada con ruido y voz solapada con música-, 1 modelo para voz de banda estrecha –voz telefónica-, 1 modelo para audios enlatados –por ejemplo anuncios- y un modelo para música. Cada modelo es representado por un GMM diagonal de 64 componentes.

Una técnica más sencilla pero con buenos resultados puede ser encontrada en [5].Además la gran ventaja de esta aproximación es que no es necesario un banco de datos externo. Esta técnica está basada en el reconocimiento de voz y música basado en características espectrales. A continuación podemos observar tres espectrogramas que corresponden a voz, música y ruido:

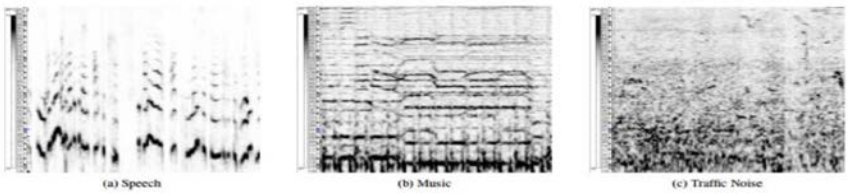

Figura 2-2: Comparación de espectrogramas para voz, música y ruido. Extraída de [5]

De esta manera podemos apreciar que, en esta representación, las señales de voz suelen mostrar patrones relativos a la presencia de varios armónicos, debidos al funcionamiento propio del tracto vocal. Es importante tener en cuenta también que, los armónicos, son sostenidos durante un corto periodo de tiempo en el cual varían ligeramente en frecuencia.

En contraposición a la señal de voz, se puede observar que la música se caracteriza por trayectorias horizontales en la zona baja del espectro y, debido a la naturaleza incorrelada del ruido, en su espectrograma no se observa ningún patrón de interés. Es de esta manera cómo podemos discriminar la señal de voz de la música o el ruido.

El objetivo por tanto sería capturar las trayectorias de los armónicos que varían ligeramente en frecuencia, al contrario de la voz cantada o los instrumentos musicales que mantienen la nota. Este hecho da como resultado una alta correlación comparando el espectro de dos tramas cercanas. Para tener también en cuenta la curvatura de las trayectorias de los armónicos se deberá calcular la correlación cruzada entre una trama X_t

y $X_{t+offset}$, que estimará el grado de correlación entre versiones desplazadas de estos vectores.

$$R_{xy}(l) = \sum_i x_i y_{i+l}$$

donde x e y son dos vectores de longitud N, y l es lag y tomará valores entre –N y +N.

Para el estudio de nuestro caso, los vectores de entrada son tramas en el dominio temporal y el lag corresponde a un desplazamiento en frecuencia. Se define entonces r_{xcorr} como la máxima correlación cruzada en un rango de lags.

$$r_{xcorr}(X_t, X_{t+offset}) = \max(R_{X_t, X_{t+offset}}(l))$$

donde $l \in [-l_{max}, l_{max}]$ marca el desplazamiento en el eje de frecuencias. Se define también el caso particular en el cual $l = 0$ y, por tanto:

$$r(X_t, X_{t+offset}) = R_{X_t, X_{t+offset}}(0)$$

Con el objetivo de obtener un indicador de presencia de voz, se define un valor tal que r_{xcorr}-r, que cumplirá que, para señales en las que solo aparece música, la correlación cruzada tendrá máximo en $l = 0$ y, por lo tanto, el valor de r_{xcorr}-$r = R(0)$-$r = r$-$r = 0$. Mientras que para señales que presenten armónicos la correlación tendrá un máximo en $l \neq 0$ y, por tanto, r_{xcorr}-r tendrá un valor estrictamente mayor que cero.

Sin embargo, para escenarios no ideales, solapamientos por ejemplo, este análisis se convertirá simplemente en un problema de optimización de umbrales de decisión. La siguiente figura muestra resultados derivados de este análisis para dos casos: transición de música a voz hablada y transición de música a voz cantada.

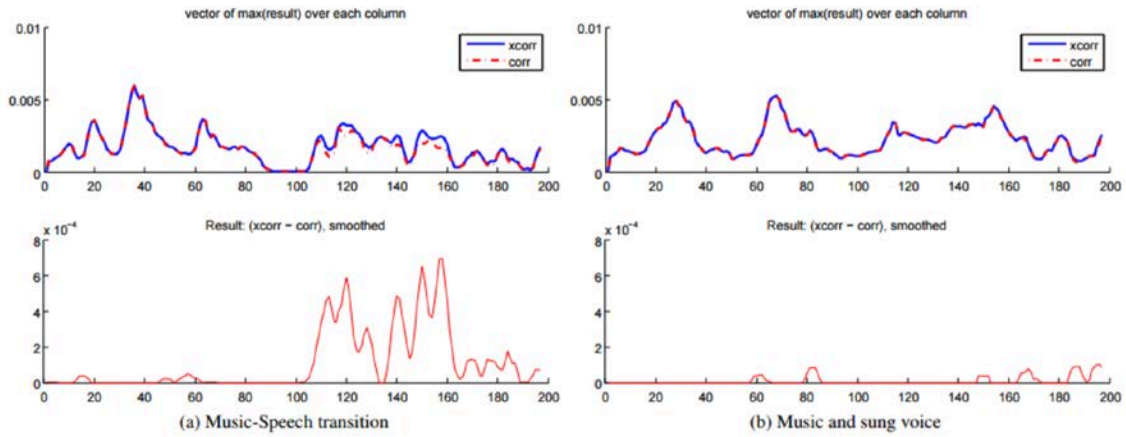

Figura 2-3: Comparación de las características espectrales de (a) transición de música a voz hablada y (b) transición de música a voz cantada. Extraída de [5]

Este método requiere un eje de frecuencias en escala logarítmica, ya que de esta manera, los armónicos aparecen en desplazamientos constantes dependientes de su armónico fundamental.

3 Entorno experimental

3.1 Base de datos

3.1.1 Creación de una base de datos

La primera etapa de este proyecto ha consistido en la elaboración de una base de datos que permitiera un estudio en un entorno controlado del sistema a estudiar. Proceso que requirió la puesta en común del trabajo de cinco integrantes del grupo ATVS-UAM.

Esta base de datos debía tener varias características importantes a tener en cuenta:

- Debía contar con programas de estructuras similares.
- Debía ser grabada en un intervalo de tiempo similar, de esta manera, los locutores y publicidades no debían sufrir grandes cambios a priori.
- Debían ser programas accesibles en con buenas calidades de audio.
- Debían ser lo más completos posibles –voz des estudio/voz telefónica, presencia de menciones y anuncios enlatados, multitud de locutores...-.

Por estos motivos se tomó la decisión que los programas más adecuados para esta base de datos eran aquellos que nos combinaban de mejor manera estas características y, por ello, nos decantamos mayoritariamente por programas de tertulia política que, en general, cubren todas nuestras necesidades.

Sin embargo, se añadió un programa recientemente a la base de datos que era un poco distinto. Este programa de música aporta unas dificultades, y a la vez oportunidades de prueba, a nuestra base de datos muy interesantes desde el punto de vista práctico. Al ser un programa de música, cobraba especial interés, en nuestro caso concreto, el análisis del VAD, para que nos distinga correctamente las zonas de música con las zonas de voz hablada.

Finalmente los programas etiquetados, horarios de grabación y etiquetado fueron:

Programa	Cadena	Emisión	Horario grabación	Etiquetado
Más de uno	Ondacero	06:00 – 12:00	08:30 – 09:30	08:30 – 09:00
Julia en la Onda	Ondacero	16:00 – 19:00	18:00 – 19:00	18:00 – 18:30
Hoy por hoy	SER	06:00 – 12:00	09:00 – 10:00	09:30 – 10:00
La Mañana	Cope	06:00 – 12:00	10:00 – 11:00	10:00 – 10:30

Tabla 3-1: Detalle programas de la Base de datos

La elección del momento óptimo de etiquetado corría a cargo del etiquetador, pero con el criterio de que fuese la media hora más completa posible. Este etiquetado fue realizado con los programas durante diez días seguidos –desde el 25 de mayo de 2015 hasta el 5 de junio de 2015-.

La base de datos por tanto consta de un total de 50 horas de programas de audio grabados, de las cuales 25 están etiquetadas.

El detalle de los datos en esta base de datos se muestra en la tabla a continuación:

	Archivo de audio	Duración sin anuncios	Duración anuncios
Más de uno	20150525.wav	27:47	02:12
	20150527.wav	28:58	01:02
	20150528.wav	28:59	01:02
	20150529.wav	27:55	02:04
	20150601.wav	28:58	01:01
	20150602.wav	29:34	00:58
	20150603.wav	28:39	01:36
	20150604.wav	27:34	02:27
	20150605.wav	29:04	00:58
	20150608.wav	28:52	01:11
Julia en la onda	20150525.wav	20:49	09:12
	20150526.wav	24:08	05:53
	20150527.wav	26:08	04:38
	20150528.wav	24:49	03:59
	20150529.wav	24:07	06:01
	20150601.wav	25:42	04:41
	20150602.wav	22:51	07:00
	20150603.wav	23:03	06:31
	20150604.wav	24:16	05:36
	20150605.wav	21:58	06:20

Hoy por hoy	20150525.wav	27:18	02:18
	20150526.wav	28:38	02:20
	20150527.wav	27:40	02:23
	20150528.wav	26:38	02:13
	20150529.wav	27:04	02:22
	20150601.wav	25:06	02:25
	20150602.wav	28:09	01:46
	20150603.wav	25:32	04:22
	20150604.wav	17:26	03:09
	20150605.wav	21:13	00:42
La mañana	20150525.wav	21:44	07:23
	20150526.wav	23:00	06:55
	20150527.wav	20:52	07:33
	20150528.wav	22:08	06:54
	20150529.wav	22:50	05:55
	20150601.wav	22:13	07:44
	20150602.wav	23:02	07:13
	20150603.wav	22:00	07:22
	20150604.wav	22:39	07:23
	20150605.wav	22:50	06:50

Tabla 3-2: Detalle duración archivos de la base de datos.

3.1.2 Etiquetado

El etiquetado de los programas de audio se ha realizado a través del entorno *Wavesurfer*, siguiendo las indicaciones dadas por Doroteo Torre en un tutorial suministrado tras la primera reunión conjunta al respecto de la base de datos.

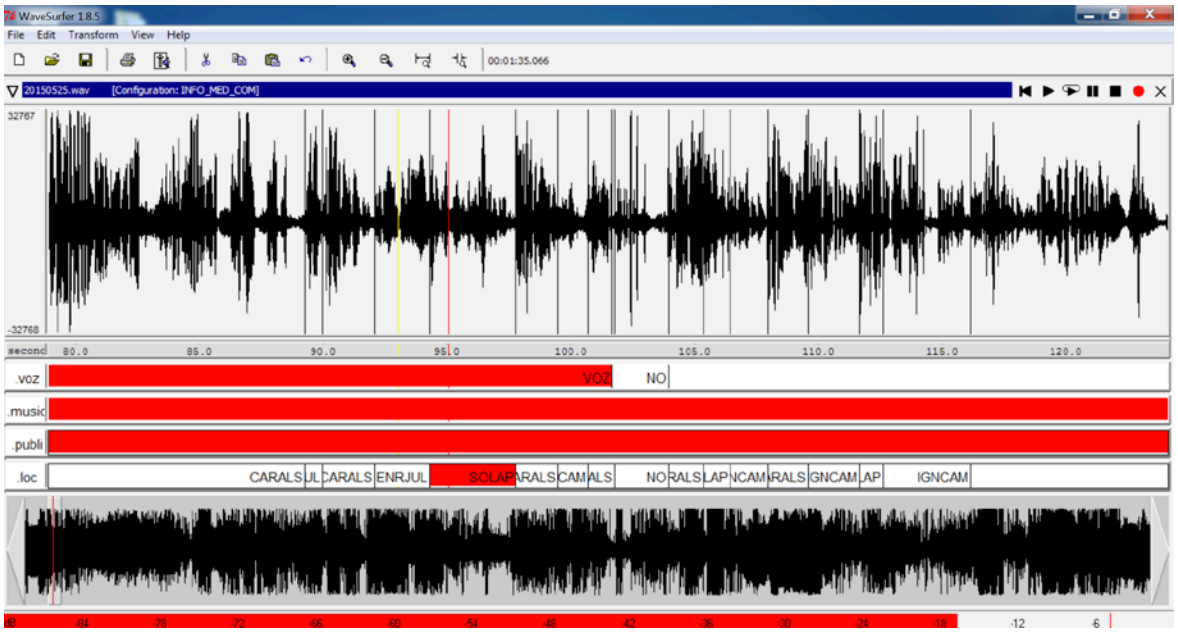

Figura 3-1: Herramienta de etiquetado de Base de Datos de audio "Wavesurfer"

Siguiendo esta guía llegamos a un etiquetado con cuatro niveles:

- "VOZ"/ "NO" / "VOZ_TEL" –refiriéndose a voz telefónica-, por lo tanto la etiqueta "VOZ" tan solo hace referencia a voz con calidad de estudio –de banda ancha-.

- "MUSICA" / "NO".

- "AN_<MARCA>_<PRODUCTO>" –denotando un anuncio enlatado de la marca y producto indicados- / "ME_<MARCA>_<PRODUCTO> -denotando una mención hecha durante el programa al producto de la marca indicados- / "NO".

- Etiquetado de locutores. Los locutores se etiquetarán de varias formas:
    - En primer lugar, los locutores conocidos serán etiquetados con las tres primeras letras de su nombre y las tres primeras letras de su apellido, así Carlos Alsina –locutor de "Más de uno" quedará como CARALS.
    - En segundo lugar, los locutores no conocidos, u ocasionales, serán etiquetados con L1, L2, L3, … , Ln.
    - En el caso de que varios locutores hablen de manera simultánea se etiquetará un solapamiento de locutores mediante SOLAP.
    - En cuarto lugar, si no se puede determinar la identidad del hablante se etiquetará con LNI –Locutor no identificable-.
    - Y, por último, en caso de que no haya locutores se etiquetará con NO.

3.2 Preparación de los datos

3.2.1 Conversión de audio

Debido al distinto origen de cada programa de radio, dado que unos se descargaban como podcast desde la web oficial y otros desde plataformas como iTunes, ha sido necesaria una normalización de los archivos de audio de manera que cada usuario que precise usar la base de datos no tenga problemas de compatibilidad con su sistema particular.

Teniendo en cuenta las necesidades de calidad, se llegó a la decisión de normalizar los audios con 16 KHz de frecuencia de muestreo, 16 bits/muestra, grabación mono y formato wav.

Esta conversión de formato se ha realizado con la herramienta "ffmpeg" de código abierto, utilizando el comando:

```
ffmpeg –i 20150525.mp3 –ac 1 –ar 16000 20150525.wav
```

3.2.2 Conversión de etiquetas

Una dificultad surgida a lo largo de este proyecto ha sido el que, para los distintos sistemas evaluados utilizan formatos distintos de etiquetado, a saber: HTK (utilizado por el ATVS), RTTM y SEG.

3.2.2.1 Formato HTK

El formato HTK es el formato utilizado para el etiquetado en la base de datos del ATVS. Es un formato ampliamente utilizado en la actualidad como formato de etiquetado de audio.

Un ejemplo de etiqueta en formato HTK sería:

```
783793474 892641028 CARALS
```

donde el primer campo es el inicio de la locución en segundos con una precisión de 7 decimales, el esgundo campo corresponde con el tiempo final de la locución y el último campo es una cadena de caracteres que identifica al locutor.

3.2.2.2 Formato RTTM

Desde 2002 a 2009, el NIST organize la evaluación de Rich Transcription. Pretendía promover avances en la tecnología de diferentes sistemas de reconocimiento del habla. Las diferentes tareas fueron categorizadas como Speech to Text Transcription (STT) o Metadata Extraction (MDE), siendo esta última la que corresponde con la tarea de diarización.

Así es como el formato RTTM se ha convertido en la referencia para las evaluaciones de la categoría de MDE.

Un ejemplo de etiquetado en RTTM sería:

```
SPEAKER session1 1 15.220 24.860 <NA> <NA> spk0 <NA>
```

Ejemplo de una etiqueta del tunning del banco de pruebas de Albayzin 2010. Donde session1 se corresponde con el nombre de la sesión estudiada, 1 corresponde al número de canales del audio en estudio, 15.220 es el inicio de la locución en segundos, 24.860 es la duración de la locución, <NA> son campos reservados, pero de momento no utilizados y, por último spk0 es una cadena que representa al locutor.

3.2.2.3 Formato SEG

Mientras que las evaluaciones de NIST están centradas en sistemas que trabajan con audio de habla inglesa hay otras evaluaciones tales como ESTER2 (2008), ETAPE (2012) y REPERE (2013) que se basan más en trabajar con sistemas de habla francesa.

El sistema LIUM estudiado en este TFG es un sistema de la universidad de Le Mans y, por tanto, ha participado en las evaluaciones francesas, obteniendo excelentes resultados.

LIUM utiliza el formato SEG que, contrariamente al RTTM, las duraciones de los segmentos de audio se dan en “features”, que, normalmente, corresponden con tramos de 10 milisegundos.

En este formato solo se documenta la actividad de cada hablante y, además, queda agrupado con los clústeres a los que corresponden.

Un ejemplo de etiqueta en SEG sería:

```
session1 1 397161 308 M S U S0
```

donde el primer campo es el nombre asignado a la sesión, el segundo es el número de canales del audio estudiado, el tercero es el tiempo de inicio –la unidad de tiempo es el features que, en nuestro caso se corresponden con 10 milisegundos-, el cuarto campo

corresponde a la duración de la locución, el quinto campo corresponde con el sexo del hablante, el sexto con la calidad de grabación de la voz –telefónica o de estudio-, el séptimo campo con el tipo de ambiente –música, discurso, etc.- y el último campo es una etiqueta que caracteriza al hablante.

3.2.2.4 Conversión SEG a RTTM

Una función de Matlab que convierte un archive SEG en uno RTTM, esta función ha sido desarrollada por Cristian Sánchez, integrante del ATVS y, es indispensable para poder utilizer la herramienta de evaluación proporcionada, que funciona con archivos RTTM.

Es importante tener en cuenta que el formato SEG es el más completo de estos tres ya que, aporta datos que los demás formatos no tienen en cuenta, como por ejemplo el sexo del hablante.

3.2.2.5 Conversión HTK a RTTM

Esta función ha sido desarrollada para que nuestra base de datos, recordemos en formato HTK, pueda ser evaluada con la herramienta de evaluación que funciona en RTTM.

Esta función por tanto recibirá unas etiquetas en HTK al que habrá que modificar los siguientes campos:

- En formato RTTM no se etiquetan los silencios, por lo que esos segmentos deberán ignorarse.
- No se etiquetará tampoco el solapamiento de locutores.
- La unidad de tiempo ha de pasarse a segundos, que es como trabaja RTTM.

3.2.2.6 Conversión RTTM a SEG y RTTM a HTK

Como comprobación de la calidad de las conversiones y, con el objetivo de poder ser ágil a la hora de convertir de cualquier formato a cualquier formato se desarrollan estas funciones que deshacen los cambios realizados.

En este caso estas funciones fueron útiles durante la evaluación de LIUM con Albayzin, evaluando la calidad de sus resultados.

En ambos casos se ha de tener en cuenta que, durante la conversión a los otros formatos, se ha perdido información irrecuperable y que, por tanto, las etiquetas no tendrán el nivel de detalle en parámetros como antes de cualquier conversión. En el caso de tiempos el cambio es perfectamente deshecho.

3.2.3 Eliminación de anuncios

Debido a que en el caso de anuncios enlatados hubo casos en los que no se etiquetó correctamente al hablante, se llegó a la conclusión de eliminar del estudio de diarización los anuncios enlatados.

Para la realización de esta tarea se desarrolló un script en Matlab que se encargaba de dividir el archivo original de audio en dos, uno con el programa sin anuncios enlatados y, otro archivo, con los anuncios concatenados uno tras otro.

Para ello la función hace uso del archivo de etiquetado de anuncios del tercer nivel de la base de datos, localiza aquellos momentos en los que la etiqueta empieza por “AN” y separa las muestras correspondientes a esa duración en otro vector.

Sin embargo, la tarea no puede quedar ahí, ya que las etiquetas de locutores que irían tras un anuncio eliminado tendrían las etiquetas mal referenciadas y, por tanto, el archivo de etiquetas ha de ser también adaptado al nuevo audio sin anuncios enlatados.

Para ello se calcula la duración del anuncio que va a ser eliminado y esta duración deberá ser restada a los tiempos de referencia, tanto de inicio como de final, de todos los locutores que llegasen tras ese anuncio.

Y no solo eso, también ha de ser tenido en cuenta que, dado que ese tiempo de anuncio había sido asignado a un solo hablante, se ha de restar la duración del anuncio al tiempo final de ese segmento, pero no al tiempo inicial.

Un error importante ocurrido en el desarrollo de esta función fue el de no tener en cuenta pequeños errores de etiquetado. Esto quiere decir que, en el momento de buscar el momento de inicio de un anuncio enlatado, se ha de tener en cuenta que es muy posible que el locutor anterior no figure todavía que ha terminado de hablar. Este hecho se debe a la enorme precisión del etiquetado, de hasta 7 decimales de segundos para el tiempo.

3.3 Medidas de rendimiento

La salida del sistema de diarización debe ser comprobada con el objetivo de comprobar la calidad de nuestro sistema. Mas, como veremos más adelante, con el objetivo de perfeccionar el sistema se deberán cambiar constantemente parámetros del programa con el objetivo de conseguir una adaptación lo más certera posible a nuestro Corpus.

De esta forma, la simple comparación entre el número de clústeres y el número de hablantes se hace insuficiente dado que puede llevar a errores de interpretación de los resultados.

3.3.1 Diarization Error Rate (DER)

El Diarization Error Rate es la medida más usada para evaluar la precisión de un sistema de diarización.

El DER puede ser definido como el tiempo total asignado incorrectamente a los hablantes en referencia al tiempo total de audio hablado, esto es, se mide en porcentaje. Debido a la ponderación temporal del DER los segmentos más largos tienen más influencia en el cálculo del DER.

El DER es resultado de la suma de tres componentes:

- False Alarm.
- Missed Speech.
- Speaker Error.

3.3.2 False Alarm (FA)

Esta componente del DER se define como el tiempo en el que se detecta voz y, sin embargo, no la hay.

Este error por tanto depende casi en exclusiva del VAD.

3.3.3 Missed Speech (MISS)

Es la componente del DER que detecta segmento sin voz a los segmentos en los que sí que existe voz.
Al igual que el FA, esta componente del DER depende también en exclusiva del VAD.

3.3.4 Speaker Error (SPKE)

Es la fracción del tiempo total asignado a un hablante incorrecto. Esta componente del DER es, por tanto la única que no depende solo del VAD. Y, por ello, es probablemente la parte más importante del DER.

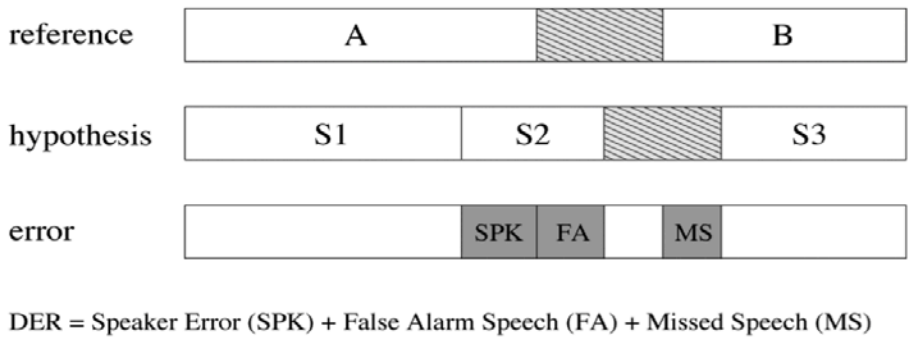

Figura 3-2: Ejemplo de cálculo de DER a partir de sus componentes.

4 Integración, pruebas y resultados

4.1 Estudio previo de herramienta LIUM

La primera etapa del estudio sin embargo ha sido el de realizar una comprobación del funcionamiento del sistema LIUM, para lo cual se organizaron una serie de pruebas con la evaluación de Albayzin 2010.

Para ello se tuvieron que optimizar los parámetros de funcionamiento del sistema, es importante tener en cuenta que este proceso es tremendamente costoso computacionalmente ya que, la única forma de realizarlo es la de ejecutar el algoritmo de manera repetida con todas las distintas combinaciones de parámetros.

En un primer estudio se realizó un barrido bastante extenso de los parámetros de entrada al sistema. Sin embargo estos resultados resultaron poco concluyentes.

Así, se obtuvieron resultados para Albayzin muy diversos entre cada sesión que era evaluada. Obteniendo no solo DER muy distintos –variando desde alrededor del 20 a más del 70%- si no que, además, estos DER óptimos resultaban para combinaciones de umbrales muy diferentes entre sí.

En el anexo de este TFG se puede observar la salida de evaluación para el último bloque del programa. Esta matriz de salida se compone de cuatro filas y tantas columnas como combinaciones de parámetros de entrada se estudian.

Observando esta salida para una sesión de Albayzin observamos:

- Que el DER mínimo obtenido para esta sesión supera el 40%.
- Para estas combinaciones de parámetros concreta –en la que se cambiaba de manera muy fina el parámetro que se observó como más importante-, la media de DER es del 54%.
- La varianza de DER es de 19,15.

De este resultado podemos ver que los valores de DER en nuestro sistema tienen una varianza grande, incluso con valores de parámetros muy cercanos entre sí. Esto implica que el sistema necesita una adaptación bastante fina a cada Corpus de estudio.

Es importante tener en cuenta que estos resultados difieren con lo esperado por varias razones:

- Este sistema obtuvo buenos resultados en varias evaluaciones francesas –ESTER 2, ETAPE y REPERE-, con valores de DER que rondaban el 25%.

- Al contrario de lo esperado, la componente de DER más importante ha sido por una elección incorrecta de hablante –SPKE-, con una media de 42%.

- En esta base de datos, se detectan con bastante precisión los segmentos de audio con voz/no voz/música, dando una media de error del 12%.

Estos resultados hicieron que en una primera fase de nuestro análisis sobre el corpus de Audio Broadcast nos centrásemos principalmente en la correcta elección del último parámetro –responsable de la asignación de hablantes de un clúster a otro-.

La razón para que estos resultados no fueran todo lo buenos que se esperaban después de las evaluaciones previas estudiadas, radica en la dificultad de la base de datos de Albayzin, consta de audios extremadamente largos –más de tres horas- y con audio muy diverso entre sí. De esta manera se puede encontrar un debate parlamentario juntado a unas noticias, a una llamada telefónica o a una emisión de noticias.

4.2 Entrenamiento

Una vez adaptado todo el sistema para su correcto funcionamiento se procede a estudiar el sistema con la base de datos de Radio-ATVS. Para el correcto estudio de la base de datos se hace indispensable una correcta elección de los parámetros de entrada que calibrarán el grado de actuación de los algoritmos de cada una de las fases.

4.2.1 Umbrales

Hay cuatro umbrales de decisión que se pueden entrenar el sistema LIUM:

- El primero fija el umbral para la segmentación del algoritmo de segmentación BIC. En nuestro estudio lo hemos hecho variar entre 1,5 y 2,5.

- El segundo umbral fija el umbral de decisión para el clustering del algoritmo BIC. En nuestro estudio lo hemos hecho variar entre 2,5 y 3,5.

- El tercer umbral fija el umbral de penalización para el algoritmo de realineamiento de Viterbi. En nuestro estudio le hemos dado los valores de 250 y 300.

- Por último el último parámetro fija los umbrales para el último clustering, que corresponde con los valores de CLR y NCLR que distinguirán los distintos clústeres. En nuestro estudio le hemos dado valores entre 1 y 3, con un paso entre los distintos valores de tan solo 0,1.

Este barrido por los distintos valores de cada uno de los parámetros de entrada resulta en una gran cantidad de archivos de diarización para cada sesión de audio usada como entrenamiento.

4.2.2 Etapas de segmentación

Mediante un script en Matlab se juntaron los resultados de cada sesión de audio y se estudiaron los valores de los umbrales que hacían el DER lo más bajo posible.

Dicho script ejecuta una herramienta de evaluación externa –mdeval-, disponible en el conjunto de archivos de la evaluación Albayzin 2010. Dicha herramienta trabaja en perl y realiza la comparación oportuna entre la segmentación del Ground Truth y la segmentación a evaluar.

Posteriormente se juntan todos los resultados obtenidos para cada programa en dos partes: la primera parte, una matriz de 4 filas, tantas columnas como archivos se estén evaluando y una tercera dimensión de valor 11 que divide los resultados para cada etapa del sistema LIUM. Y, la segunda parte un cell que contiene, ordenado por cada columna, el nombre del archivo –con su conjunto de parámetros- correspondiente al resultado de la primera matriz.

Dichas dimensiones se corresponden con:

- Show.i.seg: segmentación inicial, separados en segmentos de dos segundos.
- Show.sms.seg: segmentación tras el análisis de Speech/Music/Silence. Por tanto esta etapa se corresponde con la salida del VAD.
- Show.s.seg: segmentación basada en GLR, los segmentos se hacen más pequeños.
- Show.l.seg: segmentación tras el clustering lineal en el que, recordemos, solo se juntan aquellos clústeres similares que sean adyacentes.
- Show.h.seg: segmentación tras la etapa de clustering jerárquico.
- Show.adj.seg: segmentación tras el ajuste de los bordes de las locuciones.
- Show.flt.seg: segmentación que se produce tras el show.sms.seg, que distingue los locutores, dejando ya atrás la separación entre voz, música y silencio.
- Show.spl.seg: segmentación tras separar aquellos clústeres más largos de 20 segundos.
- Show.g.seg: el sexo del hablante y la calidad de grabación de la voz son detectadas.
- Show.c.seg: segmentación final con el clustering final del sistema que utiliza los criterios CLR y NCLR.

4.2.3 Ejecución del entrenamiento

Una vez organizado todos los elementos necesarios se realiza el entrenamiento lanzando las distintas ejecuciones del programa LIUM con cada combinación de parámetros seleccionada.

En este punto del proyecto se debió decidir el conjunto de audios destinados a entrenamiento del programa, así como el conjunto de audios destinados a la posterior evaluación del mismo.

Se llegó a la conclusión de:

- Para tres de los programas se utilizarían 7 de los 10 archivos etiquetados como conjunto de entrenamiento. Quedando 3 archivos por programa como conjunto de evaluación.

- Para el programa restante –que se seleccionó Más de Uno por sus completas características para este fin- se utilizaría la totalidad de archivos de audio como conjunto de archivos de evaluación.

Es importante tener en cuenta que dicha ejecución resulta bastante lenta, ya que, el tiempo de ejecución de LIUM, tal y como veremos más adelante, depende de los valores de los umbrales que tenga el programa.

Y no solo eso, se debe tener en cuenta el volumen de pruebas ejecutadas, obteniendo para cada batería de pruebas cientos de archivos multiplicados por el número de segmentaciones distintas que se hace para cada ejecución.

De esta manera, se realizaron 5 barridos de análisis de umbrales con más de 600 archivos de segmentación para cada sesión de cada programa utilizado como entrenamiento.

Cada barrido nos permite afinar cada umbral de decisión, de tal forma que, nuestro análisis sea lo más certero posible a la hora de elegir la combinación de parámetros óptima para nuestra base de datos. Durante el ajuste de umbrales se detecta que el sistema es especialmente sensible al ajuste del último umbral, es probable que por esa razón sea la que LIUM recomienda ajustar de manera más fina, pero también llama la atención que, el sistema por defecto no utiliza dicha etapa, hay que especificarlo con la bandera –doCEClustering.

Tras el conjunto de barridos realizados se realiza el estudio para observar los parámetros óptimos para cada programa, con el criterio de obtener un DER final lo más bajo posible. Se llega a la conclusión de que los valores de umbral óptimos para nuestra base de datos son:

- 2,0 para el primer umbral.
- 3,0 para el segundo umbral.
- 250 para el tercer umbral.

- 2,0 para el cuarto umbral.

Dichos umbrales obtienen resultados de DER para cada sesión de alrededor del 15% y, lo que es también muy importante, no hay grandes saltos en ningún caso –esto quiere decir que no se ha detectado ningún caso en el que esta combinación de parámetros obtuviese un DER llamativamente alto, pese a que la media estudiada pudiera ser muy baja-.

4.3 Evaluación

4.3.1 DER final

Una vez elegido la combinación óptima de parámetros para la base de datos, se ejecuta LIUM con esos parámetros al conjunto de archivos de evaluación. Es importante tener en cuenta que finalmente, se han obtenido 19 archivos como evaluación frente a los 21 que componen el corpus de entrenamiento.

Los resultados obtenidos han sido:

Programa La Mañana de la cadena COPE:

Archivo	DER	MISS	FA	SPKE
20150603	8,57%	0,40%	4,60%	3,50%
20150604	23,12%	0,30%	6,10%	16,80%
20150605	14,09%	0,90%	4,30%	8,90%

Tabla 4-1: Resultados para La Mañana de Cope

Programa Julia en la Onda:

Archivo	DER	MISS	FA	SPKE
20150603	8,54%	0,10%	3,20%	5,30%
20150604	16,47%	0,20%	3,00%	13,20%
20150605	8,72%	0,50%	2,80%	5,40%

Tabla 4-2: Resultados para Julia en la Onda de Ondacero.

Programa Hoy por Hoy de Cadena SER:

Archivo	DER	MISS	FA	SPKE
20150603	25,95%	0,10%	17,40%	8,40%
20150604	41,20%	23,50%	11,50%	6,30%
20150605	31,20%	0,10%	23,00%	8,10%

Tabla 4-3: Resultados para Hoy por Hoy.

Programa Más de Uno de la cadena Ondacero:

Archivo	DER	MISS	FA	SPKE
20150525	8,80%	1,70%	2,60%	4,50%
20150527	11,28%	1,10%	4,70%	5,40%
20150528	24,42%	0,30%	5,10%	18,90%
20150529	11,55%	0,50%	4,20%	6,80%
20150601	13,97%	0,60%	6,20%	7,20%
20150602	8,79%	0,30%	2%	6,40%
20150603	10,38%	0,70%	2,80%	6,90%
20150604	14,48%	0,80%	4,80%	8,90%
20150605	15,01%	1,80%	2,50%	10,70%
20150608	6,32%	0,90%	2,20%	3,30%

Tabla 4-4: Resultados para Más de Uno.

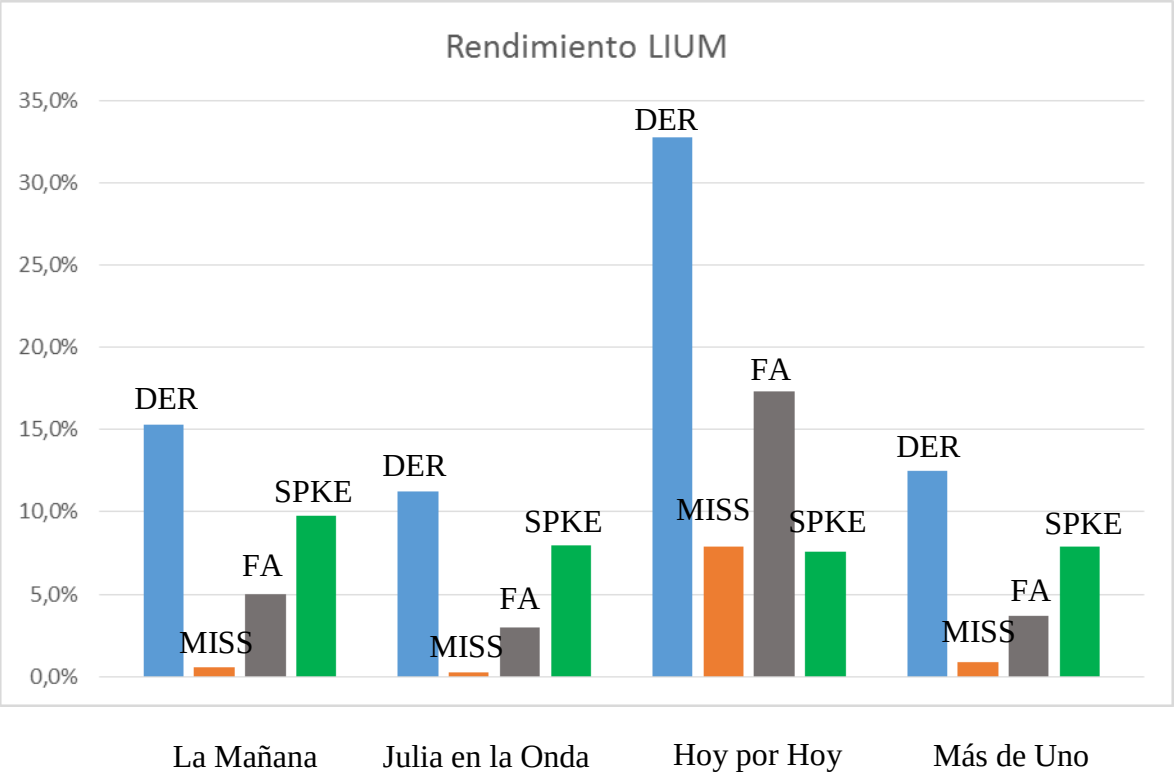

Figura 4-1: Resultados Base de Datos ATVS.

Se puede observar unos resultados realmente buenos para casi todos los programas pertenecientes al conjunto de archivos de evaluación. Sin embargo, es importante tener en cuenta que para uno de los programas se han alcanzado valores de DER mucho más altos que para el resto de programas.

Es probable que estos resultados se deban a un segmento importante de dicho programa en el que se reproduce música. Por tanto, es muy probable que los altos niveles de FA y MISS tengan origen en un mal funcionamiento del bloque VAD del sistema y, por lo tanto, una mejora a proponer al sistema sería el uso de un sistema experto alternativo externo a LIUM.

4.3.2 DER por bloque del sistema

Por último, se procede a hacer un estudio de la contribución de cada etapa del sistema al resultado final del DER.

Veamos un ejemplo de salida de un archivo:

	DER
Segmentación inicial (i)	83,96%
Speech/Music/Silence (sms)	71,97%
Small Segments (s)	103,98%
Clustering lineal (l)	84,49%
Clustering jerárquico (h)	42,77%
Algoritmo Viterbi (d)	43,51%
Ajuste de bordes (adj)	43,6%
Filtrado de hablantes de acuerdo a sms (flt)	40%
División de segmentos mayores a 20 seg. (spl)	40%
Detección de sexo y ancho de banda (g)	40%
Segmentación final (c)	13,97%

Tabla 4-5: Resultados DER por bloque.

En este ejemplo se puede apreciar la importancia de ciertas etapas del sistema. Es distinguible que, incluso, hay etapas en las que el DER resultado empeora. Este hecho se debe a que en las primeras etapas tan solo se están preparando los datos para su posterior agrupamiento y, por tanto, su objetivo no es el de conseguir unos buenos resultados de DER por sí mismos. Es por este motivo por el que tan solo tiene sentido analizar los resultados de DER a partir la etapa del clustering jerárquico.

Es de esta manera como distinguimos las dos etapas fundamentales para el correcto funcionamiento de la diarización. En primer lugar, por orden de aparición en el sistema, la etapa de clustering jerárquico reduce, en media, a la mitad el DER de la etapa que le precedía. Esta conclusión era, en parte, esperada, ya que es la primera etapa en la que se juntan en un mismo clúster segmentos que muy probablemente sean del mismo hablante independientemente de los segmentos con los que linden.

La segunda etapa fundamental es, tal como pudimos observar cuando ajustamos los umbrales a la entrada del sistema, el último clustering que da lugar a la segmentación final. Este clustering se encarga de poner en común las aportaciones del clustering jerárquico previo y de las etapas que le preceden y, además, se ajusta enormemente en base a pruebas de entrenamiento en el corpus bajo estudio.

Cabe destacar que, si bien la aportación del clustering jerárquico es de enorme importancia, es esta última etapa la que se encarga de reducir el DER entre 25 y 30 puntos porcentuales en un contexto en el que, no olvidemos, tendrá ya mucha información muy similar y, distinguir entre una y otra puede no ser trivial.

Por último, cabe destacar que, tal y como vemos en este ejemplo, es posible obtener DER por encima del 100%, esto se debe a la ecuación con la que se calcula el DER. Dicha ecuación se referencia al tiempo total de speaker y no al tiempo total de audio, por lo que, en casos con mucho error de FA y MISS es posible este escenario.

4.4 Tiempo de ejecución del sistema

El proceso de diarización es un proceso costoso desde el punto de vista computacional. Se deben hacer multitud de comprobaciones, cálculos, divisiones de señales de audio, etc. Y, por tanto, sería de esperar que los tiempos de ejecución de esta tarea fuesen elevados.

Es importante, de cara a la utilización de la herramienta en tiempo real, sea cual sea la finalidad de su utilización –etapa de preprocesado por ejemplo-, que esta tarea no solo sea certera, sino que además sea lo más rápida posible de cara a no ser un problema para el funcionamiento de otros sistemas conectados.

Durante la ejecución de este trabajo hemos podido observar que, los tiempos de ejecución del sistema dependen en gran parte de los parámetros de entrada. Sin embargo, también se ha observado que los parámetros que arrojaban mejores resultados eran también notablemente buenos en los tiempos de ejecución.

Las pruebas realizadas se han llevado a cabo sobre el servidor disponible en el laboratorio, contando con las siguientes características:

- 24 CPUs Intel® Xeon® E5645 a 2,4 GHz.
- 6 Núcleos de procesamiento por CPU.
- 16 GB de RAM.

Bajo este marco de trabajo se ha observado, para la base de datos Radio-ATVS que el tiempo de ejecución para un fichero tipo es:

Etapa de segmentación	Tiempo de ejecución
Initial Segmentation	12 segundos
Speech/Music/Silence	0,5 segundos
Linear clustering	5 segundos

Hierarchical Clustering	3 segundos
Viterbi	16 segundos
Adjust boundaries	1 segundo
Speaker Filtering	15,5 segundos
Long segments division	0,5 segundos
Gender detection	15 segundos
Final segmentation	2 minutos, 34 segundos
TOTAL	3 minutos, 42 segundos

Tabla 4-6: Tiempo de ejecución del sistema.

Estos resultados se corresponden con la ejecución del archivo 20150602 del programa "Más de Uno" con una duración de 29 minutos y 34 segundos.

Este resultado, por tanto, promete la posibilidad de utilización de este sistema de diarización en tiempo real. Sin embargo, es importante mencionar que, durante las pruebas realizadas con los audios de Albayzin 2010, se observaron tiempos de ejecución, en proporción mucho más altos que los indicados para ATVS-Radio. Este hecho se debe a que la propia longitud del archivo bajo análisis hace necesario el uso de etapas intermedias de segmentación en el clustering final. Este hecho hacía que se tuviese que repetir la última etapa –que resulta ser la más costosa- en determinados archivos hasta en cientos de veces, lo que multiplicaba enormemente su tiempo total de ejecución.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

Este Trabajo Fin de Grado perseguía dos objetivos fundamentales: por un lado y principalmente, la creación de una herramienta que nos permitiera diarizar audio broadcast español. Y, por otro lado, la creación de una base de datos con el tipo de audio que nos interesaba para el desarrollo y evaluación de dicha herramienta.

- Se ha realizado un estudio previo de la tarea de diarización. Comprobando el estado del arte a fecha de realización del proyecto y adquiriendo las herramientas necesarias para su posterior estudio.

- Se ha realizado un estudio de la herramienta LIUM, elegida por ser ganadora de las pruebas ESTER2 de diarización, para posteriormente ser evaluada con las pruebas de Albayzin 2010, llegando a resultados poco alentadores.

- Se ha creado una base de datos acorde a los datos que se deseaban estudiar con la herramienta seleccionada.

- Se han creado las funciones de Matlab necesarias para un correcto funcionamiento de nuestro corpus –en unos formatos determinados- en nuestro sistema de estudio –que trabaja con formatos distintos-.

- Se ha optimizado el funcionamiento del sistema bajo estudio para nuestra base de datos, mediante la realización de pruebas exhaustivas.

- Se ha realizado un estudio de los resultados arrojados por el sistema. Llegando a datos muy prometedores para el funcionamiento del mismo.

5.2 Trabajo futuro

- Un primer paso para el futuro sería el de la ampliación de la base de datos desarrollada en este TFG. Sería importante que dicha ampliación contase con audio de distintas naturalezas y orígenes. Un buen comienzo sería el de añadir audios de noticiarios de televisión, donde el audio está muy controlado, pero se añaden segmentos de noticias donde puede haber menos control.

- A lo largo de este trabajo se ha observado la importancia capital del sistema de detección de voz, por lo que sería interesante el probar a sustituir el sistema de LIUM por sistemas dedicados en estado del arte actual.

- Sería interesante alternar bases de datos como la actual, en español, con bases de datos en otros idiomas de diversos orígenes. Extrapolando en primer lugar a idiomas de origen común –por ejemplo, las demás lenguas cooficiales españolas- e ir

ampliando esa diversidad con idiomas de creciente diversidad con el español –por ejemplo empezando con idiomas de origen latino para luego extender el estudio a anglosajonas-.

- Por último, y dado que hemos observado que el último clustering es el que mayor impacto tiene en el DER, sería interesante realizar un estudio aislado de dicha etapa y plantear mejoras a su funcionamiento.

Referencias

[1] Claude Barras, Xuan Zhu, Sylvain Meignier and Jean-Luc Gauvain, “Multistage speaker diarization of broadcast news”. Audio, Speech and Language Processing, IEEE Transactions on 14(5):1505-1512, 2006.

[2] Gerald Friedland, “Speaker Diarization”, International Computer Science Institute, 2012.

[3] Jean-Luc Gauvain and Chin-Hui Lee. “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov Chains”. Digital signal Processing, 2000.

[4] Sylvain Meignier and Teva Merlin, “LIUM spkdiarization: an open source toolkit for diarization”. CMU SPUD Workshop, volume 2010, 2010.

[5] Benjamín García Naranjo, “Segmentación en audio Broadcast”, Universidad Autónoma de Madrid, 2016.

[6] Sue E. Tranter and Douglas A. Reynolds, “An overview of automatic speaker diarization systems”. Audio, Speech and Language Processing, IEEE Transactions on 14(5):1557-1565.

[7] Mickael Rouvier, Grégor Dupuy, Paul Gay, Elie Khoury, Teva Merlin and Sylvain Meignier, “An open source state of the art toolbox for broadcast news diarization”, LUNAM Université, LIUM, Le Mans, France.

[8] Matthew A Siegler, Uday Jain, Bhiksha Raj and Richard M Stern, “Automatic segmentation, classification and clustering of broadcast news audio”, In Proc. DARPA speech recognition workshop, volume 1997, 1997.

[9] Herbert Gish, M-H Siu and Robin Rohlicek, “Segregation of speaker for speech recognition and speaker identification”. In icassp, pages 873-876. IEEE, 1991.

[10] Thomas Rossing, “Springer Handbook of Acoustics”, springer, 2008.

[11] Chistopher M Bishop, “Pattern recognition and machine learning”, springer, 2006.

[12] Nikki Mirghafori anc Chuck Wooters. “Nuts and flakes: A study of data characteristics in speaker diarization”. In Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, volume 1, 2006.

[13] Xavier Anguera Miro, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland and Oriol Vinyals, “Speaker diarization: A review of recent research”. Audio, Speech and Language Processing, IEEE Transactions 20(2):356-370, 2012.

[14] Douglas A Reynolds, Thomas F Quateri and Robert B Dunn, “Speaker verification using adapted Gaussian mixture models”. Digital Signal processing, 10(1):19-41, 2000.

Glosario

AHC	Agglomerative Hierarchical Clustering
MFCC	Mel-Frequency Cepstral Coefficients
FFT	Fast Fourier Transform
DCT	Discrete Cosine Transform
DER	Diarization Error Rate
FA	False Alarm
MISS	Missed Speaker
SPKE	Speaker Error
HMM	Hidden Markov Model
GMM	Gaussian Mixture Model
UBM	Universal Background Model
CLR	Cross Likelihood Ratio
NCLR	Normalized Cross Likelihood Ratio
SAD	Speech Activity Detector
VAD	Voice Activity Detector

Anexos

Ejemplo de Matriz DER obtenida para la salida de un bloque de LIUM con Albayzin.

DER_Matrix_session23(:,:,2) =

Columns 1 through 8

62.3400	62.1600	61.0900	60.4300	59.1800	58.6400	57.9200	61.1900
4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000
7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000
50.3000	50.1000	49.0000	48.4000	47.1000	46.6000	45.9000	49.1000

Columns 9 through 16

60.3700	60.0200	59.2100	58.9400	58.2700	57.8900	61.4600	61.2500
4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000
7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000
48.3000	48.0000	47.2000	46.9000	46.2000	45.8000	49.4000	49.2000

Columns 17 through 24

54.7100	54.7100	52.1900	51.8200	51.4400	58.6700	57.5500	55.7200
4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000
7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000
42.7000	42.7000	40.1000	39.8000	39.4000	46.6000	45.5000	43.7000

Columns 25 through 32

55.5600	53.3700	52.9100	52.4400	57.1400	56.0100	51.8900	51.7400
4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000
7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000
43.5000	41.3000	40.9000	40.4000	45.1000	44.0000	39.8000	39.7000

Columns 33 through 40

50.7600	50.2900	49.8200	53.4900	53.0000	52.4900	48.9700	47.8900
4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000
7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000
38.7000	38.2000	37.8000	41.4000	40.9000	40.4000	36.9000	35.8000

Columns 41 through 48

47.5200	47.0000	57.9400	57.9400	54.9300	54.0000	52.1600	51.6100
4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000
7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000
35.5000	35.0000	45.9000	45.9000	42.9000	41.9000	40.1000	39.6000

Columns 49 through 56

51.0800	55.4000	54.4500	53.6100	52.6400	51.6100	51.0300	50.7100
4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000
7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000
39.0000	43.3000	42.4000	41.6000	40.6000	39.6000	39.0000	38.7000

Columns 57 through 63

51.7500	51.7500	50.9100	48.3100	47.9000	46.0800	45.8000
4.4000	4.4000	4.4000	4.4000	4.4000	4.4000	4.4000
7.7000	7.7000	7.7000	7.7000	7.7000	7.7000	7.7000
39.7000	39.7000	38.9000	36.3000	35.8000	34.0000	33.8000